REMARKS

Claims 1-36 are pending in the Application.

The examiner is thanked for the performance of a thorough search. Each issue raised in the Office Action mailed January 28, 2004 is addressed hereinafter.

I. REJECTION BASED ON 35 U.S.C. §102(e)

The Office Action has rejected Claims 1-36 under 35 U.S.C. 102(e) as being anticipated by McManis (U.S. Pat. No. 5,757,914).

Applicant respectfully disagrees.

Applicant does not believe that the Office Action has established a proper rejection under 35 U.S.C. 102(e) and again cites the following:

> In a proper rejection under § 102(e) the cited reference must show each and every claimed feature in the same combination as arranged in the claim. See Lewmar Marine, Inc. v. Barient, Inc., 827 F.2d 744, 747-48, 3 USPQ2d 1766, 1768 (Fed. Cir. 1987). If even a single element or limitation is missing from the reference, anticipation is not found. Connell v. Sears, Roebuck & Co., 722 F.2d 1542, 1548, 220 USPQ 193, 198 (Fed. Cir. 1983).

Claims 1, 13, and 25 appear as follows:

> 1. A method of securely invoking an access control function, the method comprising the steps of:
>
> receiving a digital signature for the access control function;

generating a mapping of the access control function to the digital signature;

determining that the digital signature is mapped to the access control function based on the mapping when execution of the access control function is requested;

determining whether an executable element matches the access control function based on the digital signature;

executing the executable element only when the executable element matches the access control function; and

wherein a particular class defines an implementation of the access control function.

13. A computer-readable medium carrying one or more sequences of one or more instructions for securely invoking an access control function, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

receiving a digital signature for the access control function;

generating a mapping of the access control function to the digital signature;

determining that the digital signature is mapped to the access control function based on the mapping when execution of the access control function is requested;

determining whether an executable element matches the access control function based on the digital signature;

executing the executable element only when the executable element matches the access control function; and

wherein a particular class defines an implementation of the access control function.

25. An access control system, comprising:

a processor;

a memory coupled to the processor;

a first mapping that maps each of a set of access control functions to a digital signature of that access control function;

the processor configured to retrieve an executable element in response to a request to execute a first access control function;

the processor configured to determine whether the executable element matches the first access control function based on the digital signature;

the processor configured to execute the executable element when the executable element matches the first access control function; and

wherein the set of access control functions are each implemented in a class.

The Office Action has misinterpreted the elements in the Claims. McManis does not disclose a system wherein a particular class defines an implementation of the access control function as claimed in the invention. Under the Response To Amendment section, the Office Action states:

> "The Examiner asserts that McManis discloses wherein a particular class defines an implementation of the access control function, because McManis discloses methods, specifically method (128)(see col. 3, lines 11-12)."

Col. 3 lines 8-18 state:

> "As shown in FIG. 1, in a preferred embodiment of the invention each application program object instance includes an object header 122, at least one digital signature 124, at least one embedded public encryption key 126 and a main application procedure 128 (often called a method). Each method or procedure 128 includes at least one verifier procedure call instruction 130 and instructions 132 for responding to a verification denial message received in response to the verifier procedure call, such as instructions for aborting execution of the procedure."

McManis defines a method as a main application procedure. Fig. 1 shows two main application procedures (methods) 128-A and 128-B. McManis makes not mention of a class defining an implementation of an access control function. McManis is only concerned with **when** his method calls the verifier. McManis states: "Each method or procedure 128 includes at least one verifier procedure call instruction 130 and instructions 132 for responding to a

verification denial message received in response to the verifier procedure call, such as instructions for aborting execution of the procedure".

There is no correlation between McManis' method and the claimed invention's classes of access control functions. Therefore, McManis does not contemplate a system wherein a particular class defines an implementation of the access control function as claimed in the invention.

The Office Action further states:

> "The Examiner asserts that functions in the class typically manipulate the member variables. These member variables are referred to methods of a class. Thus, the methods such as method (128)(see col. 3, lines 17-25), are part of the class. The Examiner asserts that these methods determine what the objects of the class do."

It is unclear as to how the Office Action reasons the correlation between member variables and methods, as the circular reasoning stated above is unreadable. It is unclear where member variables apply. Further, member variables are not methods. Additionally, the Office Action does not refer to classes and objects as defined in the Specification.

Col. 3, lines 17-25 state:

> "The main application A procedure (128-A) in the first program module furthermore includes a procedure call 134 to an executable procedure (e.g., the main application B procedure 128-B) in the second procedure module. The procedure call 130-A to the

program module verifier is logically positioned in the first program module so as to be executed prior to execution of the procedure call 134 to the second program module."

Here, McManis is merely stating the premise of his invention, *i.e.*, that the procedure call to the program module verifier in the main application procedure (method) precedes the procedure call to the second program module. There is no correlation between the cited passage from McManis and the element of a particular class defining an implementation of the access control function as cited in Claims 1, 13, and 25. McManis teaches away from Claims 1, 13, and 25 by teaching that a single program module verifier is called before other program modules are called. Col. 3, lines 26-38 state:

"The procedure call 130-B to the program module verifier is logically positioned in the second program module immediately after the entry point to each executable procedure 128-B in the second program module so as to be executed prior to execution of each such procedure 128-B. More generally, in other embodiments of the invention the procedure call 130-B to the program module verifier is logically positioned in the second program module (and more generally, in all program modules that will be called by other program modules) prior to the completion point in each executable procedure in the second program module so as prevent completion of the execution of each such procedure if verification of the calling program is denied by the verifier."

Further, the Office Action is using a keyword match in McManis to support an assertion that is inaccurate. The Office Action states:

"The Examiner asserts that the verifier is an object (see col. 4, lines 7-8). The Examiner asserts that an object is an instance of a class."

Col. 4, lines 7-12 state:

"More specifically, the verifier, which is preferably a distinct trusted object (or alternately a trusted system service procedure) receives the request message from procedure A (step 206), and decodes (step 208) a digital signature embedded in program module B using a public key provided by the calling procedure (i.e., procedure A)."

A trusted object as used in McManis is used in a different context as what is defined in the Specification. The mere use of the word "object" does not define an instantiation of a class as defined in the Specification.

McManis therefore does not teach every aspect of the claimed invention.

Applicant asserts that the rejection under 35 U.S.C. 102(e) is improper and the finality of the Office Action is premature. Applicant respectfully requests that the Examiner withdraw the finality of the Office Action.

Claims 1, 13, and 25 are allowable. Claims 2-12, and 14-24, and 26-36 are dependent upon Claims 1, 13, and 25, respectively, and are allowable. Applicant respectfully requests that the Examiner withdraw the rejection under 35 U.S.C. 102(e).

II. CONCLUSIONS & MISCELLANEOUS

For the reasons set forth above, Applicant respectfully submits that all pending claims are patentable over the art of record, including the art cited but not applied. Accordingly, allowance of all claims is hereby respectfully solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: March 29, 2004

Kirk D. Wong
Reg. No. 43,284

1600 Willow Street
San Jose, CA 95125
Telephone No.: (408) 414-1080 ext.214
Facsimile No.: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Non-Fee Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

on March 29, 2004 by